

**YAPILAN İŞ :** Daha önce yapılan ScrollView çalışmasının içindeki ListView yapısına dışarıdan eleman ekleme

**NOT :** Daha önceki raporda kod satırları anlatıldığı için sadece eklenti yapılan yerler rapora eklenecektir.

```
211 ▼ Rectangle {
212     id: namerect
213     width: 80; x:0
214     height: 40
215     anchors.bottom: name2rect.top
216     anchors.left: parent.left
217     border.width: 1
218     border.color: 'black'
219 ▼ Text{
220     id : nametext
221     text: " Coin Name : "
222     anchors.centerIn: parent
223     font.bold: true
224     font.pointSize: 8
225 }
226
227 }
228
229 ▼ Rectangle{
230     id:namerect1
231     width: 120; x:80
232     height: 40
233     anchors.bottom:name2rect1.top
234     anchors.left: namerect.right
235     border.width: 1
236     border.color: 'black'
237 ▼ TextArea{
238     id:namelist
239     font.bold: true
240     font.pointSize: 8
241 }
242 }
243
244 ▼ Rectangle {
245     id: name2rect
246     width: 80; x:0
247     height: 40
248     anchors.bottom: parent.bottom
249     anchors.left: parent.left
250     border.width: 1
251     border.color: 'black'
252 ▼ Text{
253     id : name2text
254     text: "Parity : "
255     anchors.centerIn: parent
256     font.bold: true
257     font.pointSize: 8
258 }
259
260 }
261
```

**NOT 1 :** Burada yapılan çalışma listview yapısının alt kısmında gerekli bilgilerin girileceği TextArea'lar oluşturulup bunları listview'e eklemektir.

**NOT 2 :** Eklenecek bilgiler önce hangi bilgi eklenecekse o bilginin yazılı olduğu bir Rectangle onun sağında ise bilginin yazılacağı Textarea'nın bulunacağı Rectangle yazılarak düzenlenmiştir.

**NOT 3 :** Bu düzen program ekranında en alt bölüme yerleştirilmiştir. İstenildiği takdirde daha farklı şekillerde yapılabilir.

**NOT 4 :** Tüm Rectangle'lar aynı mantıkla yapılmış olup tek fark yerleştirildikleri yerdir.

```

262 ▼
263
264
265
266
267
268
269
270 ▼
271
272
273
274
275
276
277 ▼
278
279
280
281
282
283
284
285 ▼
286
287
288
289
290
291
292
293
294 ▼
295
296
297
298
299
300
301
302 ▼
303
304
305
306
307
308

```

```

Rectangle{
    id:name2rect1
    width: 120; x:80
    height: 40
    anchors.bottom: parent.bottom
    anchors.left: name2rect.right
    border.width: 1
    border.color: 'black'
    TextArea{
        id:name2list
        font.bold: true
        font.pointSize: 8
    }
}
Rectangle{
    id: name3rect
    width: 80; x:200
    height: 40
    anchors.bottom: parent.bottom
    anchors.left: name2rect1.right
    border.width: 1
    border.color: 'black'
    Text{
        id : name3text
        text: "Hacim : "
        anchors.centerIn: parent
        font.bold: true
        font.pointSize: 8
    }
}
Rectangle{
    id:name3rect1
    width: 120; x:180
    height: 40
    anchors.bottom: parent.bottom
    anchors.left: name3rect.right
    border.width: 1
    border.color: 'black'
    TextArea{
        id:name3list
        font.bold: true
        font.pointSize: 8
    }
}

```

```

309 ▼
310
311
312
313
314
315
316
317 ▼
318
319
320
321
322
323
324
325
326 ▼
327
328
329
330
331
332
333
334 ▼
335
336
337
338
339
340
341 ▼
342
343
344
345
346
347
348
349 ▼
350
351
352
353
354
355
356
357

```

```

Rectangle{
    id: name4rect
    width: 80; x:200
    height: 40
    anchors.bottom: name3rect.top
    anchors.left: namerect1.right
    border.width: 1
    border.color: 'black'
    Text{
        id : name4text
        text: "Birim : "
        anchors.centerIn: parent
        font.bold: true
        font.pointSize: 8
    }
}
Rectangle{
    id:name4rect1
    width: 120; x:180
    height: 40
    anchors.bottom: name3rect1.top
    anchors.left: name4rect.right
    border.width: 1
    border.color: 'black'
    TextArea{
        id:name4list
        font.bold: true
        font.pointSize: 8
    }
}
Rectangle{
    id: name5rect
    width: 80
    height: 40
    anchors.bottom: parent.bottom
    anchors.left: name4rect1.right
    border.width: 1
    border.color: 'black'
    Text{
        id : name5text
        text: "Yüzde : "
        anchors.centerIn: parent
        font.bold: true
        font.pointSize: 8
    }
}

```

```

358 ▼
359     id:name5rect1
360     width: 120
361     height: 40
362     anchors.bottom: parent.bottom
363     anchors.left: name5rect.right
364     border.width: 1
365     border.color: 'black'
366 ▼
367     TextArea{
368         id:name5list
369         font.bold: true
370         font.pointSize: 8
371     }
372 }
373 ▼
374 Rectangle{
375     id: name6rect
376     width: 80
377     height: 40
378     anchors.bottom: name5rect.top
379     anchors.left: name4rect1.right
380     border.width: 1
381     border.color: 'black'
382     Text{
383         id : name6text
384         text: "Miktar : "
385         anchors.centerIn: parent
386         font.bold: true
387         font.pointSize: 8
388     }
389 }
390 ▼
391 Rectangle{
392     id:name6rect1
393     width: 120; x:180
394     height: 40
395     anchors.bottom: name5rect1.top
396     anchors.left: name6rect.right
397     border.width: 1
398     border.color: 'black'
399     TextField{
400         id:name6list
401         font.bold: true
402         font.pointSize: 8
403     }
404 }

```

```

405 Button{
406     id:addlistview
407     width: 150
408     height: 80
409     anchors.bottom: parent.bottom
410     anchors.right: parent.right
411     text: "ADD LIST"
412     onClicked: {
413         myListView.model.append({name:name1list.text,name2:name2list.text,hacim:name3list.text,birim:name4list.text,yüzde:name5list.text,miktar:name6list.text})
414         name1list.clear()
415         name2list.clear()
416         name3list.clear()
417         name4list.clear()
418         name5list.clear()
419         name6list.clear()
420     }
421 }
422 }
423 }
424 }
425 }








```

**NOT 5 :** 413. Satırda bulunan myListView adlı listview içine append fonksiyonu kullanılarak TextArea içinde yazılanları listview'e eklenmektedir. Buradaki önemli detay append fonksiyonun içinde yazılanlar listview içinde kullanılan variable'lerden ilgili olanla eşleştirilmeli veya yeni bir variable oluşturulup listview içindeki bir elementin text komutunda eşitlenmelidir.

**NOT 6 :** clear fonksiyonu listview'e eklendikten sonra TextArealar içinde yazılanların silinmesi için kullanılmıştır.

Scroll					
<b>STEEM</b> /ETH	Birim : 0.000610	+40.55%	Miktar : 12	SUB	ADD
HACİM = 3682	Price : \$9.39				
<b>ARK</b> /BTC	Birim : 0.00004986	+34.36%	Miktar : 23	SUB	ADD
HACİM = 540	Price : \$24.51				
<b>ARDR</b> /USDT	Birim : 0.40293	+29.41%	Miktar : 76	SUB	ADD
HACİM = 11.12M	Price : \$3.37				
<b>DENT</b> /USDT	Birim : 0.114235	+28.64%	Miktar : 47	SUB	ADD
HACİM = 507.28M	Price : \$0.0136618				
<b>GTO</b> /USDT	Birim : 0.05118	+20.65%	Miktar : 89	SUB	ADD
HACİM = 15.49M	Price : \$0.427950				
<b>FIL</b> /USDT	Birim : 146.5713	+17.07%	Miktar : 95	SUB	ADD
HACİM = 472.50M	Price : \$1225.58				
<b>ZRX</b> /USDT	Birim : 1.7577	+14.94%	Miktar : 33	SUB	ADD
HACİM = 47.16M	Price : \$14.70				
Coin Name :		Birim :		Miktar :	
Parity :		Hacim :		Yüzde :	
					ADD LIST

**Program Çalıştırıldığında**

 STEEM /ETH	Birim : 0.000610	+40.55%	Miktar : 12	SUB	ADD
HACİM = 3682	Price : \$9.39				
 ARK /BTC	Birim : 0.00004986	+34.36%	Miktar : 23	SUB	ADD
HACİM = 540	Price : \$24.51				
 ARDR /USD	Birim : 0.40293	+29.41%	Miktar : 76	SUB	ADD
HACİM = 11.12M	Price : \$3.37				
 DENT /USD	Birim : 0.114235	+28.64%	Miktar : 47	SUB	ADD
HACİM = 507.28M	Price : \$0.0136618				
 GTO /USD	Birim : 0.05118	+20.65%	Miktar : 89	SUB	ADD
HACİM = 15.49M	Price : \$0.427950				
 FIL /USD	Birim : 146.5713	+17.07%	Miktar : 95	SUB	ADD
HACİM = 472.50M	Price : \$1225.58				
 ZRX /USD	Birim : 1.7577	+14.94%	Miktar : 33	SUB	ADD
HACİM = 47.16M	Price : \$14.70				

## Veriler Yazıldığında

Screenshot of a web browser window showing a trading interface for various cryptocurrencies. The interface lists coins like STEEM, ARK, ARDR, DENT, GTO, FIL, ZRX, and AAA with their current prices, price changes, and quantities. At the bottom, there is a table with columns for Coin Name, Parity, Birim, Hacim, Miktar, and Yüzde. An 'ADD LIST' button is visible on the right.

<div> <div>STEEM /ETH</div> <div> <div>Birim : 0.000610</div> <div>+40.55%</div> <div>Miktar : 12</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 3682</div> <div>Price : \$9.39</div> </div>																	
<div> <div>ARK /BTC</div> <div> <div>Birim : 0.00004986</div> <div>+34.36%</div> <div>Miktar : 23</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 540</div> <div>Price : \$24.51</div> </div>																	
<div> <div>ARDR /USDT</div> <div> <div>Birim : 0.40293</div> <div>+29.41%</div> <div>Miktar : 76</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 11.12M</div> <div>Price : \$3.37</div> </div>																	
<div> <div>DENT /USDT</div> <div> <div>Birim : 0.114235</div> <div>+28.64%</div> <div>Miktar : 47</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 507.28M</div> <div>Price : \$0.0136618</div> </div>																	
<div> <div>GTO /USDT</div> <div> <div>Birim : 0.05118</div> <div>+20.65%</div> <div>Miktar : 89</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 15.49M</div> <div>Price : \$0.427950</div> </div>																	
<div> <div>FIL /USDT</div> <div> <div>Birim : 146.5713</div> <div>+17.07%</div> <div>Miktar : 95</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 472.50M</div> <div>Price : \$1225.58</div> </div>																	
<div> <div>ZRX /USDT</div> <div> <div>Birim : 1.7577</div> <div>+14.94%</div> <div>Miktar : 33</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 47.16M</div> <div>Price : \$14.70</div> </div>																	
<div> <div>AAA /BBB</div> <div> <div>Birim : 8.8745</div> <div>+27.27%</div> <div>Miktar : 66</div> <div>SUB</div> <div>ADD</div> </div> <div>HACİM = 54.54M</div> <div>Price : undefined</div> </div>																	
<div> <table> <tr> <td>Coin Name :</td> <td></td> <td>Birim :</td> <td></td> <td>Miktar :</td> <td></td> </tr> <tr> <td>Parity :</td> <td></td> <td>Hacim :</td> <td></td> <td>Yüzde :</td> <td></td> </tr> </table> <div>ADD LIST</div> </div>						Coin Name :		Birim :		Miktar :		Parity :		Hacim :		Yüzde :	
Coin Name :		Birim :		Miktar :													
Parity :		Hacim :		Yüzde :													

## Veri ListView'e Eklendiğinde

## YAPILAN İŞ : Q\_PROPERTY Yapısını Kullanılarak C++ 'dan QML 'e Sinyal Yollama(Veri Yollama)

**NOT :** İlk olarak kod yapıları anlatılacak ardından yapım aşamaları anlatılacaktır.

```
1  #ifndef MYPerson_H
2  #define MYPerson_H
3
4  #include <QObject>
5
6  class MyPerson : public QObject
7  {
8      Q_OBJECT
9      Q_PROPERTY(QString name READ name WRITE set_name NOTIFY name_changed)
10     Q_PROPERTY(int age READ age WRITE set_age NOTIFY age_changed)
11
12 public:
13     explicit MyPerson(QObject *parent = nullptr);
14     MyPerson(const QString &name, int age, QObject *parent = nullptr);
15     QString name()const
16     {
17         return m_name;
18     }
19
20     int age()const
21     {
22         return m_age;
23     }
24
25     void set_name(const QString &name);
26     void set_age(int age);
27
28 signals:
29     void name_changed();
30     void age_changed();
31 private:
32     QString m_name;
33     int m_age;
34 };
35
36 #endif // MYPerson_H
```

**GÖRSEL 1 -> myperson.h**

**NOT 1 :** Private olarak m\_age ve m\_name variable'ları tanımlanmıştır.

**NOT 2 :** Signals içinde Qml dosyalarını hangi fonksiyon ile gönderileceği yazılmıştır.

**NOT 3 :** Public içindeki QString name() ve int age() fonksiyonları diğer cpp dosyalarında m\_name ve m\_age variable'larını kullanmak için return etmektedir.

**NOT 4 :** Satır 14 deki kodlar main.cpp ve myperson.cpp sayfalarında anlatılacaktır.

**NOT 5 :** Satır 25 ve 26 'daki set\_name ve set\_age fonksiyonları burada tanımlanıp myperson.cpp sayfasında işlevi belirtilmiştir.

**NOT 6 :** Satır 9 'daki Q\_PROPERTY yapısı;

QString name = qml de kullanacağımız variable

READ name = header ve cpp de hangi variable'ı okuyup yollanacağını belirtir

WRITE set\_name = Gönderilecek variable'ın nerede setlendiği belirtilir.

NOTIFY name\_changed hangi sinyal ile yollanacağını belirtir.

```

1  #include "myperson.h"
2
3  ▼ MyPerson::MyPerson(QObject *parent) : QObject(parent)
4  {
5      m_age = 0;
6  }
7  ▼ MyPerson::MyPerson(const QString &name ,int age, QObject *parent)
8  {
9      m_name = name;
10     m_age = age;
11 }
12
13 ▼ void MyPerson::set_name(const QString &name)
14 {
15     if(m_name != name){
16         m_name = name;
17         emit name_changed();
18     }
19 }
20
21 ▼ void MyPerson::set_age(int age)
22 {
23     if(m_age != age){
24         m_age = age;
25         emit age_changed();
26     }
27 }
28

```

**GÖRSEL 2 -> myperson.cpp**

**NOT 7 :** myperson.h dosyasında tanımlanan set\_name ve set\_age fonksiyonları açıklanmıştır. Burada private olan m\_name qml de kullanmak için name ile eşitlenmiştir. Signals'de tanımlanan name\_changed() yapısı burada emit ile kullanılmıştır.

Emit bir variable'a erişmek için kullanılır.

**NOT 8 :** Not 7'deki mantıkla set\_age yapısı oluşturulmuştur.

**NOT 9 :** Satır 7 'de başlayan kod satırı ilk olarak main.cpp 'de MyPerson adlı bir ObjectList oluşturulmuştur. Ardından header'da satır 14 'de bu objectList'in içinde bulunacak variable'lar belirtilmiştir. Buradaki asıl mantık daha önce belirtildiği gibi set\_name ve set\_age 'de name ve age

ayarlanıyor. Burada ise bu liste bu

variable'lara erişim sağlamaktadır.

```

1  #include <QGuiApplication>
2  #include <QQmlApplicationEngine>
3  #include <QQmlContext>
4  #include <QQuickStyle>
5  #include "myperson.h"
6
7  ▼ int main(int argc, char *argv[])
8  {
9
10     QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);
11
12     QQuickStyle::setStyle("Material");
13
14     QGuiApplication app(argc, argv);
15
16     QQmlApplicationEngine engine;
17
18     QObjectList people;
19     people.append(new MyPerson("Okan",23));
20     people.append(new MyPerson("Keso",24));
21     people.append(new MyPerson("Ömer",26));
22     people.append(new MyPerson("Mehmet",48));
23     people.append(new MyPerson("Onur",30));
24     people.append(new MyPerson("Eda",28));
25     people.append(new MyPerson("Sema",33));
26
27     engine.rootContext()->setContextProperty("people",QVariant::fromValue(people));
28
29     engine.load(QUrl(QStringLiteral("qrc:/main.qml")));
30     if(engine.rootObjects().isEmpty())
31         return -1;
32     return app.exec();
33 }
34

```

**GÖRSEL 3 -> main.cpp**

**NOT 10 :** Satır 18'de people adlı bir object list oluşturulmuştur. Ardından bu object liste eleman eklemeleri yapılmıştır.

```

1  import QtQuick 2.15
2  import QtQuick.Controls 2.5
3  import QtQuick.Layouts 1.3
4
5  ApplicationWindow {
6      id:root
7      width: 640
8      height: 480
9      visible: true
10     title: qsTr("Hello World")
11
12     ListView{
13         anchors.fill: parent
14         model:people
15         delegate: Rectangle{
16             width: parent.width
17             height: 30
18             RowLayout{
19                 spacing: 20
20                 width: parent.width
21
22                 height: inc_age_button.height
23                 Text{
24
25                     text: name
26                 }
27                 Text{
28
29                     text: age
30                 }
31                 Button{
32                     id:inc_age_button
33                     text: "Age +1"
34                     onClicked: {
35                         age = age+1;
36                     }
37                 }
38                 Button{
39                     id:dec_age_button
40                     text: "Age -1"
41                     onClicked: {
42                         age = age-1;
43                     }
44                 }
45                 TextField{
46                     id:name_textfield
47                     width: root.width /5
48                     placeholderText: 'Enter Name'
49                     Button{
50                         text: "OK"
51                         width: parent.width/4
52                         anchors{
53                             right: parent.right
54                             top: parent.top
55                             bottom: parent.bottom
56                         }
57                         onClicked: {
58                             name=name_textfield.text
59                         }
60                     }
61                 }
62             }
63         }
64     }
65 }

```

**GÖRSEL 3 -> main.qml**

**NOT 11 :** Burada bir listview oluşturulmuştur. Bu listview’de model olarak main.cpp ‘de oluşturulan people kullanılmıştır.

**NOT 12 :** Ardından kullanılacak verilerin programda yazdırılması için rowlayout oluşturulmuştur. Burada text’ler içinde kullanmak için diğer cpp dosyalarında hazırlanan age ve name kullanılmıştır.

**Not 13 :** Diğer butonlarda ise age variable’ın arttırma ve azaltma özelliği eklenmiştir.

**NOT 14 :** TextField bölümünde ise text’de yazılan name değiştirilmesi yapılmıştır.

**NOT 15 :** Q\_Property yapısı temel olarak anlaşılmıştır ancak main.cpp ‘de yapılan tanımlanmalar tam olarak anlaşılmamıştır. Aynı zamanda Q\_Property’nin bu örnek dışında kullanımları da vardır. O yapılar da incelenmelidir.



## PROGRAM OLUŞTURULMASI

1. İlk olarak QT Quick Application – Empty projesi oluşturulmuştur. Ardından Add New C++ Class oluşturulmuştur. Bu Class oluşturulurken Base Class seçeneğinde QObject seçilmiştir.
2. Daha sonra header kısmında ilk olarak private variable'lar yazılmıştır. Daha sonra signaller yazılıp ardından public içinde bu private variable'ları return etmek için fonksiyonlar yazılmıştır.
3. Diğer adım ise myperson.cpp dosyasında düzenlenmek için set\_name ve set\_age fonksiyonları tanımlanıp myperson.cpp dosyasında içeriği yazılmıştır.
4. Header dosyası içinde Q\_PROPERTY yapısı oluşturulmuştur.
5. main.cpp dosyası içinde düzenlemeler yapıp QObjectList oluşturulmuştur. Daha sonra bu objectlisti kullanmak için ve variable'lara erişmek için sadece cpp dosyalarında kullanılacak myperson adlı model oluşturulmuştur. Burada erişilecek variable'lar tanımlanıp main.cpp dosyasında append fonksiyonu ile listeye elemanlar eklenmiştir.
6. main.qml dosyasında ise istenilen formatta ayarlamalar yapılmıştır.

**NOT :** Program oluşturulması bölümünde kodlar anlatılmamıştır.

**NOT :** Program görünüm düzenlenmesi tam olarak yapılmamıştır.



**GÖRSEL 4 -> Program çalıştırıldığında**

Hello World

Okan	21	AGE +1	AGE -1	aaa	...
Keso	26	AGE +1	AGE -1	Enter Name	...
Ömer	26	AGE +1	AGE -1	Enter Name	...
Mehmet	48	AGE +1	AGE -1	Enter Name	...
Onur	30	AGE +1	AGE -1	Enter Name	...
Eda	28	AGE +1	AGE -1	Enter Name	...
Sema	33	AGE +1	AGE -1	Enter Name	...

“ **GÖRSEL 5 -> Age değiştirilip textfield kısmına yazım yapıldığında**

Hello World

aaa	21	AGE +1	AGE -1	aaa	...
Keso	26	AGE +1	AGE -1	Enter Name	...
Ömer	26	AGE +1	AGE -1	Enter Name	...
Mehmet	48	AGE +1	AGE -1	Enter Name	...
Onur	30	AGE +1	AGE -1	Enter Name	...
Eda	28	AGE +1	AGE -1	Enter Name	...
Sema	33	AGE +1	AGE -1	Enter Name	...

**GÖRSEL 6 -> İsim Değiştirildiğinde**

**YAPILAN İŞ :** C++'dan QML 'e veri gönderimi için kullanılan yöntemlerden birisi olan Context Property yapısı incelenmiştir.

```
1  #ifndef BACK_H
2  #define BACK_H
3
4  #include <QObject>
5  #include <QString>
6
7  class Back : public QObject
8  {
9      Q_OBJECT
10 public:
11     explicit Back(QObject *parent = nullptr);
12
13 signals:
14     void valueChanged(QString s);
15
16 public slots:
17     void changeValue(int a);
18 };
19
20 #endif // BACK_H
```

**GÖRSEL 1 -> back.h**

**NOT 1 :** Header dosyasında Q\_PROPERTY yapısından farklı olarak slots yapısı vardır. Aslında bu konudaki temel mantık C++'dan QML 'e bir veya birden fazla veri kullanırken signali bir slot içine yerleştirip QML dosyasında bu slotu kullanmaktır. Burada da bu signal ve slot yapısı tanımlanmıştır.

```
1  #include "back.h"
2
3  Back::Back(QObject *parent) : QObject(parent)
4  {
5
6  }
7
8  void Back::changeValue(int a)
9  {
10     if(a==1){
11         valueChanged("value is 1, From c++");
12     }
13
14     else if (a==2){
15         valueChanged("value is 2, From c++");
16     }
17
18     else{
19         valueChanged("!!!!");
20     }
21 }
```

**GÖRSEL 2- > back.cpp**

**NOT 2 :** Bu kısımda back.h dosyasında slot olarak tanımladığımız changeValue fonksiyonu düzenlenmiştir. Burada change value içindeki int a 'nın değerine göre gönderilecek signal tanımlanmıştır.

```

1  #include <QGuiApplication>
2  #include <QQmlApplicationEngine>
3  #include <QQmlContext>
4
5  #include "back.h"
6
7  int main(int argc, char *argv[])
8  {
9      Back *back = new Back;
10
11      QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);
12
13      QGuiApplication app(argc, argv);
14
15      QQmlApplicationEngine engine;
16      engine.rootContext()->setContextProperty("back",back);
17      engine.load(QUrl(QStringLiteral("qrc:/main.qml")));
18      if (engine.rootObjects().isEmpty())
19          return -1;
20
21
22      return app.exec();
23 }

```

### GÖRSEL 3 -> main.cpp

**NOT 3 :** main.cpp de ilk olarak QQmlContext include edilmelidir.

**NOT 4 :** Satır 9 ' da Back adlı class'a ait bir object tanımlanmıştır.

**NOT 5 :** Satır 16 'da bu tanımlanan object ve o objecte ait olan verileri qml dosyalarında kullanmak için bu object'e bir isim belirtilmiştir. Buradaki "back" qml dosyasındaki kullanıcığımız ismi, onun yanındaki back ise satır 9 da oluşturduğumuz object'i ifade eder.

**NOT 6 :** Satır 17 ' de ise bu objecti hangi qml dosyasında kullanacağımız belirtilir.

```

1  import QtQuick 2.15
2  import QtQuick.Window 2.15
3  import QtQuick.Controls 2.5
4
5  ▼ Window {
6      width: 640
7      height: 480
8      visible: true
9      title: qsTr("Hello World")
10
11  ▼ Button{
12      id:button
13      x:47
14      y:229
15      text: "Button1"
16      antialiasing: true
17      scale: 1.3
18      onClicked: back.changeValue(1);
19  }
20  ▼ Button{
21      id:button1
22      x:163
23      y:229
24      text: "Button2"
25      antialiasing: true
26      scale: 1.3
27      onClicked: back.changeValue(2);
28  }
29  ▼ Button{
30      id:button2
31      x:279
32      y:229
33      text: "Button3"
34      antialiasing: true
35      scale: 1.3
36      onClicked: back.changeValue(3);
37  }
38  ▼ Label{
39      id:label
40      x:23
41      y:88
42      width: 355
43      height: 62
44      text: "Label"
45      horizontalAlignment: Text.AlignHCenter
46      elide: Text.ElideMiddle
47      font.capitalization: Font.MixedCase
48      font.family: "Arial"
49      font.pointSize: 20
50  ▼ Connections{
51      target:back
52      onValueChanged:label.text = s;|
53  }
54  }
55  }

```

**GÖRSEL 4 -> main.qml**

**NOT 7 :** main.qml dosyasında ise back.cpp dosyasında ayarlamaları yapılan change value fonksiyonu ve main.cpp dosyasında oluşturulan object referred 'ı kullanmaktayız.

Belirtilen öğeleri kullanmak için en basit olarak 3 tane buton oluşturulmuştur ve bu butona basıldığında bu fonksiyon çağrılmaktadır.

Çağırılma şekli ise ilk olarak referred edilen object ve ona ait olan changevalue yazılır.

**NOT 8 :** Label içinde ise çağırılan bu change value içindeki signal yazdırılır.

Burada ilk Label'ın yerleştirilmeleri yapıp ardından Connections yapısı kullanılır.

Connection içindeki target yapısı sinyal gönderimi yapan nesneyi tutar. Bu örnekte ise sinyal gönderimi yapan nesne back'tir.

Ardından label'ın içindeki text sinyal fonksiyonu olan valueChanged(QString s)

içindeki variable olan "s" 'e eşitlenmektedir.

Hello World

— □ ×

Label

Button1

Button2

Button3

**GÖRSEL 5 -> Program Çalıştırıldığında**

Hello World

— □ ×

value is 1, From c++

Button1

Button2

Button3

**GÖRSEL 6 -> Buton 1 ' e Basıldığında**

Hello World

— □ ×

value is 2, From c++

Button1

Button2

Button3

**GÖRSEL 7 -> Buton 2 'ye basıldığında**

Hello World

— □ ×

!!!!

Button1

Button2

Button3

**GÖRSEL 8 -> Buton 3 'e Basıldığında**

## YAPILAN İŞ : Q\_INVOKABLE Yapısının incelenmesi

```
1  #ifndef DATASTORE_H
2  #define DATASTORE_H
3
4  #include <QObject>
5
6  class datastore : public QObject
7  {
8      Q_OBJECT
9      Q_PROPERTY(QString message READ message WRITE setMessage NOTIFY messageChanged)
10
11  public:
12      explicit datastore(QObject *parent = 0);
13      QString message(){return msg;}
14      void setMessage(QString str);
15      Q_INVOKABLE int qInvokeExample(QString str);
16
17  signals:
18
19      void messageChanged();
20
21  public slots:
22      void callMeFromQml();
23
24  private:
25      int count ;
26      QString msg;
27
28  };
29
30 #endif // DATASTORE_H
```

### GÖRSEL 1 -> datastore.h

**NOT 1 :** İlk olarak private variable olan count ve msg tanımlanmıştır. Ardından signal olarak kullanacağımız messageChanged() fonksiyonu ve slot olarak kullanacağımız callMeFromQml() tanımlanmıştır.

**NOT 2 :** Daha sonra message ve setMessage fonksiyonları yazılmıştır. Ardından Q\_INVOKABLE yapısını kullanmak için qInvokeExample fonksiyonu tanımlanmıştır.

**NOT 3 :** setMessage , qInvokeExample ve callMeFromQml fonksiyonları datastore.cpp dosyasında yazılmıştır.

**NOT 4 :** Q\_Property yapısında main.qml yapısında kullanacağımız variable read kısmı okuması yapılacak olan variable setMessage bu message variable'ın nerede yazıldığı ve messageChange ise sinyali belirtmektedir.

```

1  #include "datastore.h"
2
3  ▼ datastore::datastore(QObject *parent) : QObject(parent), count(0), msg("0 Hits")
4  {
5
6  }
7
8  ▼ void datastore::setMessage(QString str)
9  {
10     msg = str;
11     emit messageChanged();
12 }
13
14 ▼ int datastore::qInvokeExample(QString str)
15 {
16     qDebug(str.toLatin1());
17     return count;
18 }
19
20 ▼ void datastore::callMeFromQml()
21 {
22     qDebug("Inside callMeFromQml()");
23     count++;
24     setMessage(QString("%1 hits").arg(count));
25 }
26

```

## GÖRSEL 2 -> datastore.cpp

**NOT 5 :** setMessage fonksiyonu private olarak tanımlanan msg variable'ını kullanmak için str'ye eşitlenir ve bu sinyal ile yollamak için messageChanged ile emit edilir.

**NOT 6 :** qInvokeExample fonksiyonu ile programın debug kısmında yazdırmak için qDebug(str.toLatin1()) yazılır. toLatin1 fonksiyonu ise döndürülen string tanımsız ise, döndürülen byte dizisini de tanımsız yapmak için kullanılır.

**NOT 7 :** callMeFromQml fonksiyonu ise debug kısmında verilecek message belirtilir. Ayrıca her çağrıldığında count arttırılır.



```

1  import QtQuick 2.15
2  import QtQuick.Window 2.15
3  import QtQuick.Controls 2.5
4
5  ▼ Window {
6      width: 640
7      height: 480
8      visible: true
9      title: qsTr("Hello World")
10
11  ▼ Column{
12      id:firsCol
13  ▼ Text {
14      id: txtCount
15      text: datastore.message
16  }
17  ▼ Button{
18      id:btnUpdate
19
20      text:"Just Click Me"
21      onClicked:datastore.callMeFromQml()
22  }
23  }
24
25  ▼ Column{
26      anchors.left: firsCol.right
27  ▼ Button{
28      text: "Click Me"
29  ▼ onClicked: {
30      var result = datastore.qInvokeExample("I pass whatever i want");
31  }
32  }
33  }
34  }
35  }

```

**GÖRSEL 3 -> main.qml**